

## The Problem

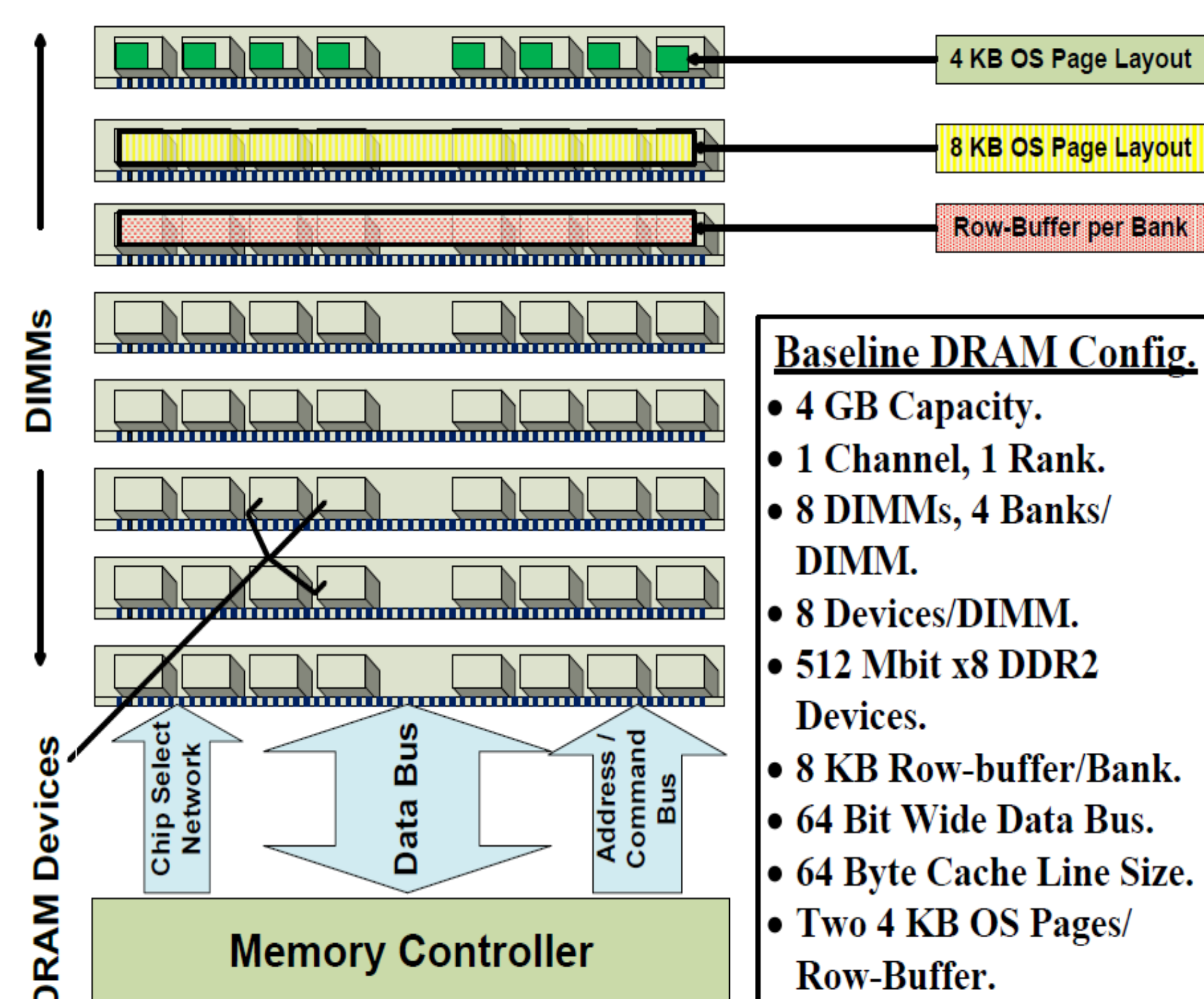
### The memory wall !!

Main memory latency still is a bottleneck – with many-core CPUs it gets worse.

### The power wall !!

Power consumed by the DRAM sub-system of a data-center server accounts for ~ 30% of the total power consumed.

## Baseline Organization



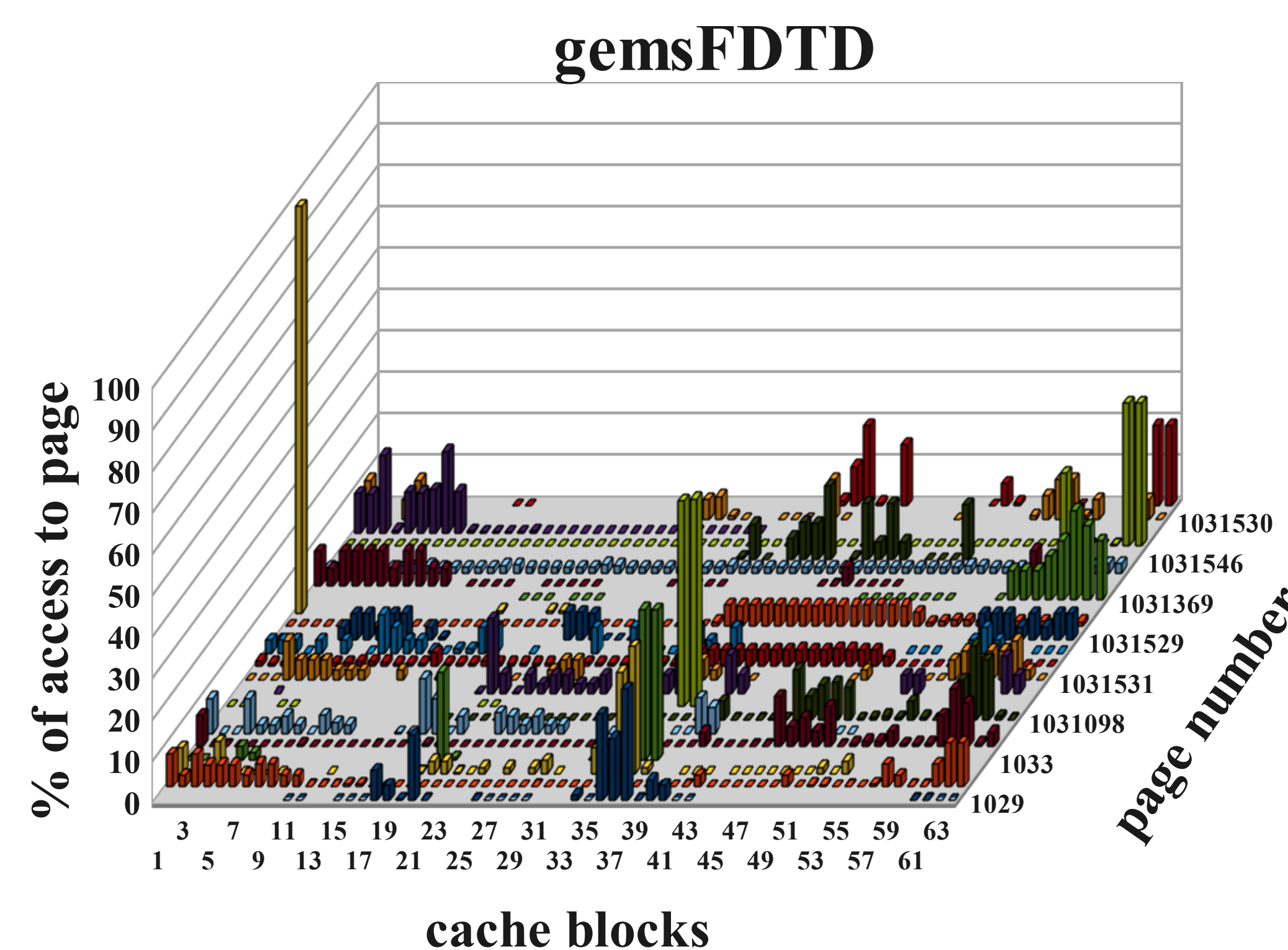
**Goal is to increase the DRAM row-buffer hit-rate to reduce power and access latency**

## Basic Idea

**Perform intelligent data placement in DRAM to maximize locality of accesses thereby promoting row-buffer reuse.**

**A key observation here is that accesses to an OS page are clustered around a few cache blocks in that page.**

## Locality of Accesses



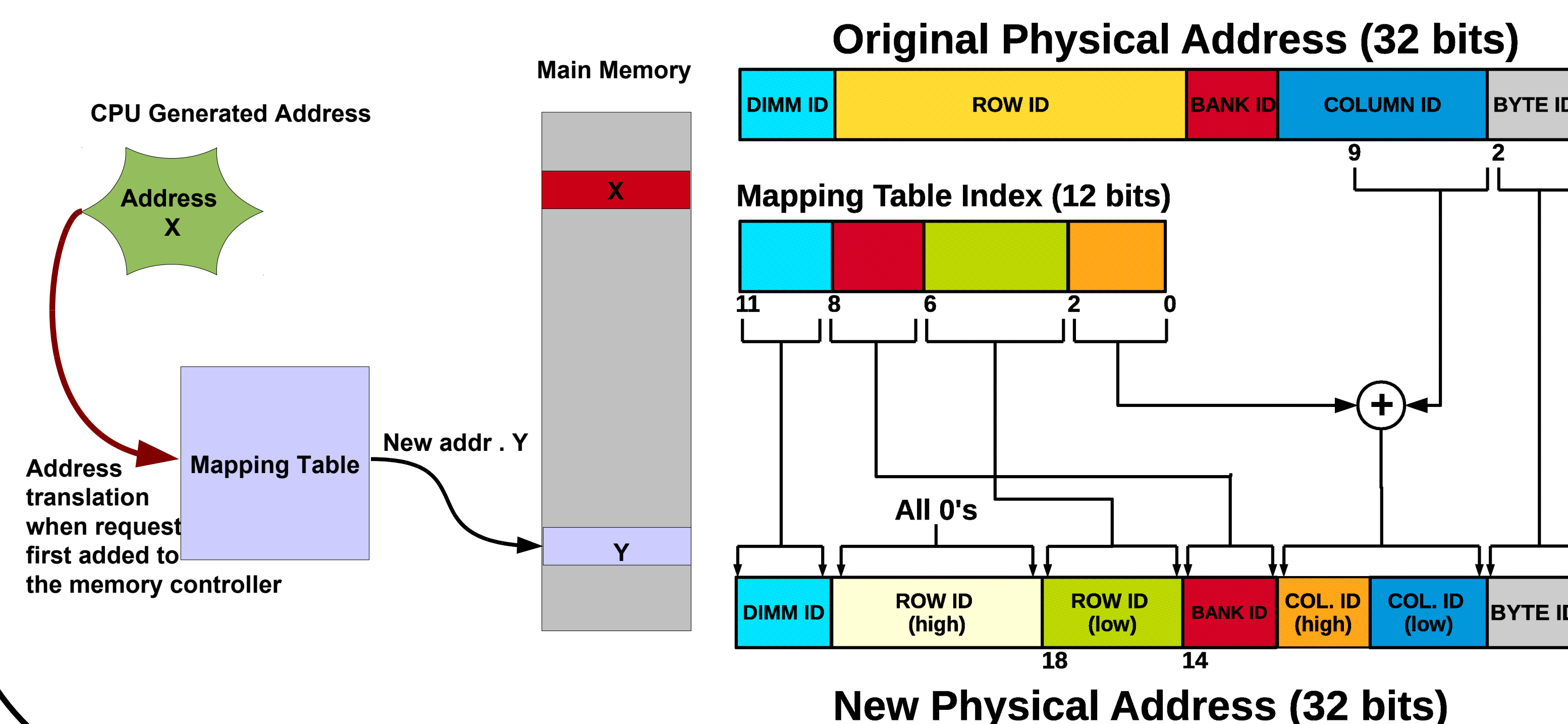
**Co-locate clusters of heavily-accessed contiguous blocks from different OS pages to improve the row-buffer utilization.**

## Reducing OS Page Size (ROPS)

- ◆ Reduce OS page size to isolate the “hot” clusters in a *micro-page*. Micro-pages are 1KB in size.
- ◆ Reserve rows in DRAM to accommodate hot micro-pages.
- ◆ Every epoch, an OS daemon selects hot micro-pages based on access counts, and selected micro-pages are migrated to the reserved rows using DRAM copies.
- ◆ OS' page table is updated and to reduce the page-table size and prevent TLB coverage drop, superpages are created from cold micro-pages.

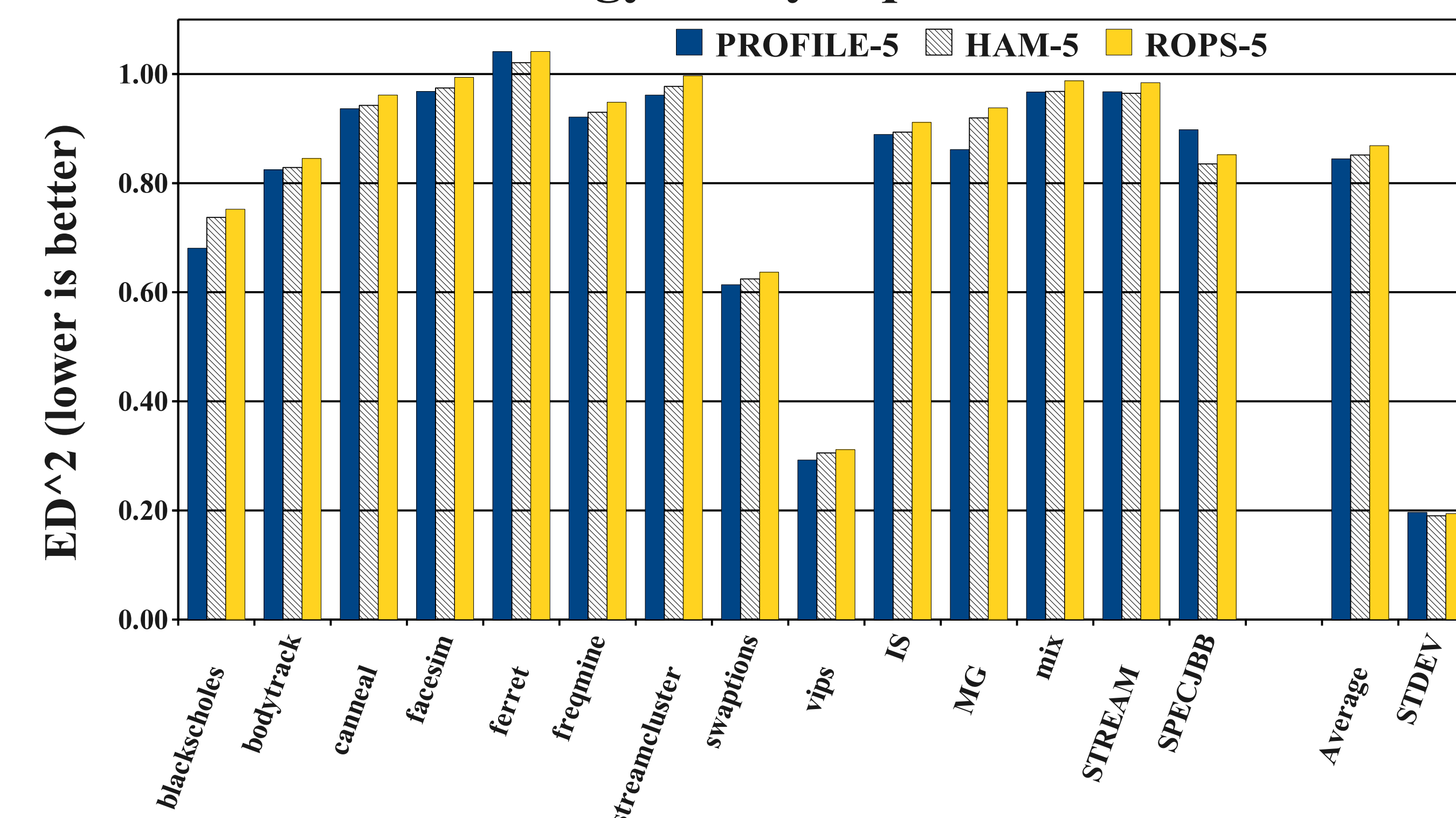
## Hardware Assisted Migration (HAM)

- ◆ New level of indirection in the address mapping scheme – the Mapping Table (MT).
- ◆ Memory controller intercepts requests to hot micro-pages and uses MT to re-direct the requests. No changes to OS' page-table.

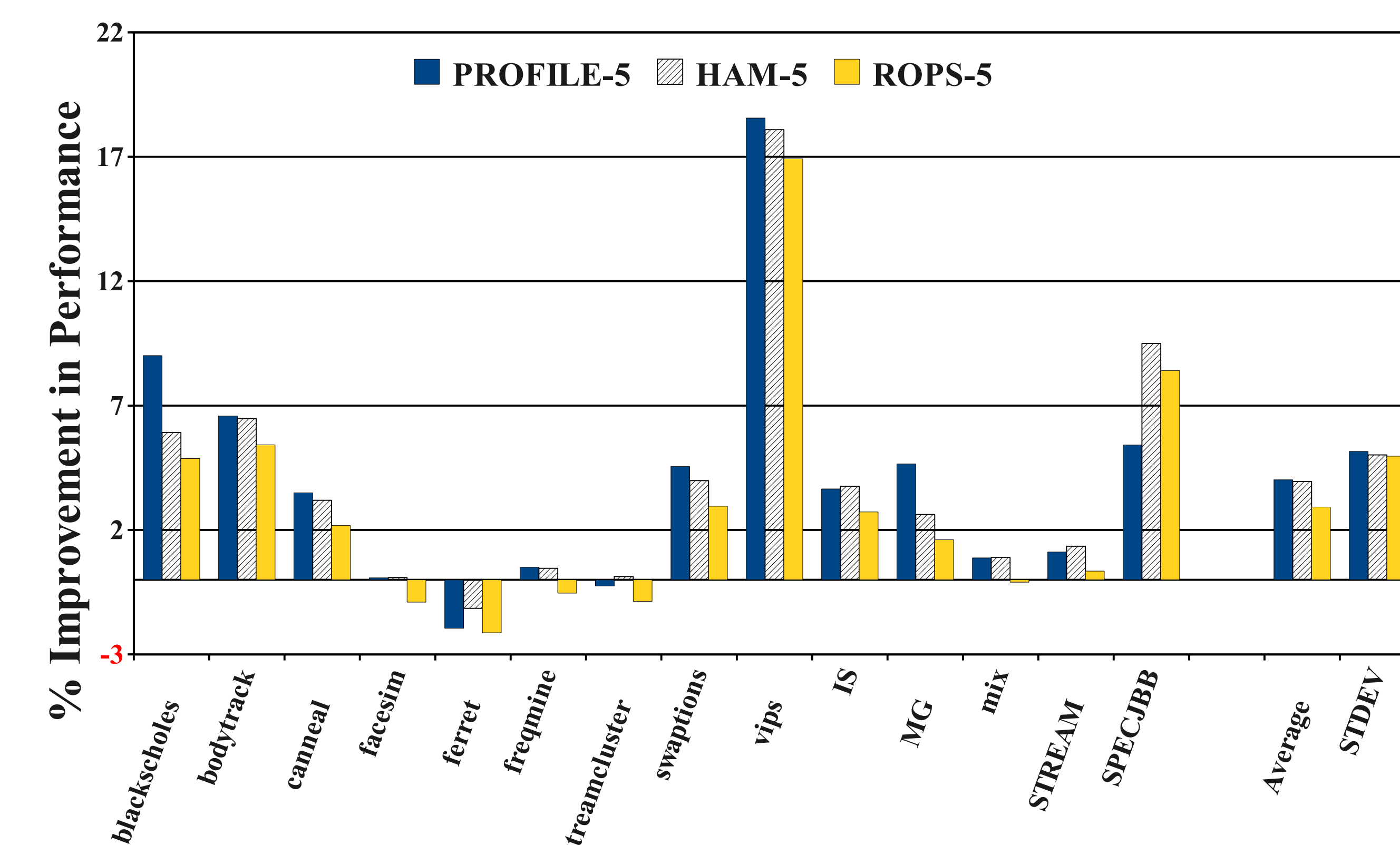


## Results

### Energy-Delay-Square



### Performance Improvement



- ◆ For these experiments, we simulated three policies: ROPS, HAM and PROFILE. Heavily accessed micro-pages are **migrated every 5 Million CPU cycles**.
- ◆ PROFILE is an oracular experiment designed to quantify an upper-bound on the performance. It quantifies the benefits if a **theoretically perfect** placement of data occurs.

## Conclusions

- ◆ On average, our best performing scheme **increases performance by 9%** (max. 18%) and **reduces memory energy consumption by 15%** (max. 70%).
- ◆ Hardware assisted migration offers better returns due to fewer overheads of TLB shutdown/misses.

*Appears at ASPLOS-2010.*

*Find out more @ <http://www.cs.utah.edu/~rajeev/pubs/asplos10.pdf>*